Properties

A property is a member that provides a flexible mechanism to read, write, or compute the value of a private field. Properties can be used as if they are public data members, but they are actually special methods called accessors. This enables data to be accessed easily and still helps promote the safety and flexibility of methods.

Properties enable a class to expose a public way of getting and setting values, while hiding implementation or verification code. A **`get`** property accessor is used to return the property value, and a **`set`** property accessor is used to assign a new value. These accessors can have different access levels. The **`value`** keyword is used to define the value being assigned by the set accessor.

Properties can be **read-write** (they have both a get and a set accessor), **read-only** (they have a get accessor but no set accessor), or **write-only** (they have a set accessor, but no get accessor). Write-only properties are rare and are most commonly used to restrict access to sensitive data. Simple properties that require no custom accessor code can be implemented either as expression body definitions or as **auto-implemented properties**.

Example:

```
using System;
using System.Globalization;

public class Person
{
   private string _firstName;
   private string _lastName;
   private DateTime _dob;

   public Person(string first, string last)
   {
      _firstName = first;
      _lastName = last;
      _dob = DateTime.Now;
   }
// Property with expression body definition
   public string Name => $"{_firstName} {_lastName}";

// auto implemented property
   public DateTime DOB {get; set;}

// read only property with get accessor body
   public int Age
   {
       get {

            return DateTime.Now.Year - DOB.Year;
       }
   }
}
```

```
public class Example
{
   public static void Main()
   {
      var person = new Person("Rohit", "Sharma");
      Console.WriteLine("Name : " + person.Name);
      Person.DOB = DateTime.ParseExact("30-04-1987", "dd-MM-yyyy",
CultureInfo.InvarientCulture);
      Console.WriteLine("Date of Birth : " + person.DOB.ToString("dd/MM/yyyy",
CultureInfo.InvarientCulture));
      Console.WriteLine("Age : " + person.Age + "Years");
   }
}
// The example displays the following output:
//    Name : Rohit Sharma
//    Date of Birth : 30/04/1987
//    Age : 34 Years
```